Marta Čertíková; Pavel Burda; Jaroslav Novotný; Jakub Šístek

Some remarks on averaging in the BDDC method

In: Jan Chleboun and Petr Přikryl and Karel Segeth and Jakub Šístek (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 6-11, 2010. Institute of Mathematics AS CR, Prague, 2010. pp. 28--34.

Persistent URL: http://dml.cz/dmlcz/702737

SOME REMARKS ON AVERAGING IN THE BDDC METHOD*

Marta Čertíková, Pavel Burda, Jaroslav Novotný, Jakub Šístek

1 Introduction

The Balancing Domain Decomposition based on Constraints (BDDC) method introduced in [1] is one of the latest domain decomposition methods. It can be understood as an improvement of the primal Neumann-Neumann domain decomposition method. As it has been recently shown in [3], a primal preconditioner of such type is determined by the choice of two operators: the injection R and the averaging E. These two operators appear also in the estimate of the condition number of the preconditioned operator (see (4) bellow).

The choice of the operator R can be formulated as the choice of continuity conditions across the interface (coarse unknowns). A lot of work has been invested into research of relations between the choice of coarse unknowns and the quality of preconditioning, and significant results were obtained (e.g. in [2, 3]).

On the other hand, the averaging operator E seems to be aside from the main effort of the investigation so far. Standard choices of E found already in [1] are arithmetic average and average weighted by diagonal entries of matrices of local problems.

In this paper, we introduce a general framework for derivation of the averaging operator, from which the standard choices are recovered by simplifications. Then, an alternative approach derived by another simplification is proposed and tested on a 2D example.

2 Reduction of the problem to the interface

Let us consider a boundary value problem with a self-adjoint operator defined on a domain $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$. If we discretize the problem by means of the standard finite element method (FEM), we arrive at the solution of a system of linear equations in the matrix form

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{K}$ is large, sparse, symmetric positive definite (SPD) matrix and $\mathbf{f}$ is the vector of the right-hand side.

*This work was supported by Czech Academy of Sciences through AV0Z10190503, by grant No. 106/08/0403 of the Czech Science Foundation and by the project of the Research Centre 1M0554. We also acknowledge fruitful conversations with Jan Mandel, University of Colorado Denver.

Let us decompose the domain Ω into N non-overlapping subdomains Ω_i, $i = 1, \dots N$. Unknowns common to at least two subdomains form the *global interface* denoted as Γ. Remaining unknowns are classified as belonging to subdomain *interiors*. The global interface Γ can be expressed as union of *local interfaces* Γ_i, $i = 1, \dots N$, containing interface unknowns involved just in subdomain Ω_i.

The first step used in many domain decomposition methods including BDDC is the reduction of the problem to the interface. Without loss of generality, suppose that unknowns are ordered so that interior unknowns form the first part and the interface unknowns form the second part of the solution vector, i.e. $\mathbf{u} = \left[\begin{array}{cc} \mathbf{u}_\mathrm{o} & \widehat{\mathbf{u}} \end{array} \right]^T$, where $\mathbf{u}_\mathrm{o}$ stands for all interior unknowns and $\widehat{\mathbf{u}}$ for unknowns at interface. Now, system (1) can be formally rewritten to block form

$$\left[\begin{array}{cc} \mathbf{K}_\mathrm{oo} & \mathbf{K}_\mathrm{or} \\ \mathbf{K}_\mathrm{ro} & \mathbf{K}_\mathrm{rr} \end{array} \right] \left[\begin{array}{c} \mathbf{u}_\mathrm{o} \\ \widehat{\mathbf{u}} \end{array} \right] = \left[\begin{array}{c} \mathbf{f}_\mathrm{o} \\ \widehat{\mathbf{f}} \end{array} \right]. \tag{2}$$

The hat symbol ($\widehat{\ }$) is used to denote global interface quantities. If we suppose the interior unknowns ordered subdomain after subdomain, then the submatrix $\mathbf{K}_\mathrm{oo}$ is block diagonal with each diagonal block corresponding to one subdomain.

After eliminating all the interior unknowns from (2), we arrive at *Schur complement problem* for the interface unknowns

$$\widehat{\mathbf{S}}\,\widehat{\mathbf{u}} = \widehat{\mathbf{g}}, \tag{3}$$

where $\widehat{\mathbf{S}} = \mathbf{K}_\mathrm{rr} - \mathbf{K}_\mathrm{ro}\mathbf{K}_\mathrm{oo}^{-1}\mathbf{K}_\mathrm{or}$ is the *Schur complement* of (2) with respect to the interface and $\widehat{\mathbf{g}} = \widehat{\mathbf{f}} - \mathbf{K}_\mathrm{ro}\mathbf{K}_\mathrm{oo}^{-1}\mathbf{f}_\mathrm{o}$ is sometimes called *condensed right-hand side*. Interior unknowns $\mathbf{u}_\mathrm{o}$ are determined by interface unknowns $\widehat{\mathbf{u}}$ via the system of equations $\mathbf{K}_\mathrm{oo}\mathbf{u}_\mathrm{o} = \mathbf{f}_\mathrm{o} - \mathbf{K}_\mathrm{or}\widehat{\mathbf{u}}$, which represents N independent subdomain problems with Dirichlet boundary condition prescribed on the interface and can be solved in parallel. The main objective represents the solution of problem (3), which is solved by the preconditioned conjugate gradient method (PCG).

3 Primal DD methods and BDDC

The main idea of the primal DD substructuring methods of Neumann-Neumann type can be expressed as splitting the given residual of PCG method to subdomains, solving subdomain problems and projecting the result back to the global domain. The primal preconditioner can be written as $M = ES^{-1}E^T$, where operator E^T represents splitting of the residual to subdomains, S^{-1} stands for solution of subdomain problems, and E represents projection of subdomain solutions back to the global problem by some averaging [3]. The condition number κ of the preconditioned operator $M\widehat{S}$ is bounded by

$$\kappa \leq ||RE||_S^2, \tag{4}$$

where operator R splits the global interface into subdomains and the energetic norm on the right-hand side is defined by the scalar product as $||u||_S^2 = \langle Su, u \rangle$. The

relationship (4) was proved in [3] assuming that $ER = I$, which means that if the problem is split into subdomains and then projected back to the whole domain, the original problem is obtained.

If we use independent subdomain problems only (no continuity conditions across the interface), the operator S is expressed by a block diagonal matrix $\mathbf{S}$ with diagonal blocks $\mathbf{S}_i$ representing local Schur complements on subdomains. Relationship between global and local problems can be expressed in matrix form as

$$\widehat{\mathbf{S}} \;=\; \mathbf{R}^{\mathrm{T}}\mathbf{S}\mathbf{R} \;=\; \sum_i {\mathbf{R}^i}^{\mathrm{T}}\mathbf{S}^i\mathbf{R}^i\,, \quad \mathbf{u} = \mathbf{R}\widehat{\mathbf{u}}\,, \quad \widehat{\mathbf{u}} = \mathbf{E}\mathbf{u}, \tag{5}$$

where $\mathbf{R}^i$ represents prolongation operator from local (subdomain) interface Γ_i to the global interface Γ and $\mathbf{E}$ performs some averaging.

The main idea of the BDDC ([1]) is to introduce a global *coarse problem* in order to achieve better preconditioning and to fix 'floating subdomains' by making their local Schur complements invertible. The matrix $\mathbf{S}$ is then positive definite, but it is not block diagonal any more, R now represents splitting of the global interface into subdomains except the coarse unknowns, and E^T distributes residual among neighbouring subdomains only in those interface unknowns which are not coarse. Thus in BDDC, only part of the global residual is split into subdomains; residual at the coarse unknowns is left undivided – it is processed by the global coarse problem.

4 Choice of the averaging operator E

We start by algebraic analysis of an elliptic problem on a domain divided into two subdomains, assuming coarse unknowns to be values at nodes only, and then generalize the results. For an illustration of this simple case see Figure 1.

4.1 Projection RE and its complement in matrix representation

Let us assume that on the interface there are m coarse nodes and n nodes which are not coarse. Suppose that nodes are ordered so that nodes that are not coarse are numbered subdomain by subdomain and the coarse nodes are the last. Then, in the simple case of two subdomains, the vectors $\widehat{\mathbf{u}}$ and $\mathbf{u}$ of values at the interface nodes and the matrices $\widehat{\mathbf{S}}$, $\mathbf{S}$ and $\mathbf{R}$ will have the following structure:

$$\widehat{\mathbf{S}} = \begin{bmatrix} \widehat{\mathbf{S}}_{\mathrm{rr}} & \widehat{\mathbf{S}}_{\mathrm{rc}} \\ \widehat{\mathbf{S}}_{\mathrm{cr}} & \widehat{\mathbf{S}}_{\mathrm{cc}} \end{bmatrix},\ \mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathrm{r}}^1 \\ \mathbf{u}_{\mathrm{r}}^2 \\ \mathbf{u}_{\mathrm{c}} \end{bmatrix},\ \mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mathrm{rr}}^1 & \mathbf{0} & \mathbf{S}_{\mathrm{rc}}^1 \\ \mathbf{0} & \mathbf{S}_{\mathrm{rr}}^2 & \mathbf{S}_{\mathrm{rc}}^2 \\ \mathbf{S}_{\mathrm{cr}}^1 & \mathbf{S}_{\mathrm{cr}}^2 & \widehat{\mathbf{S}}_{\mathrm{cc}} \end{bmatrix},\ \mathbf{R} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}, \tag{6}$$

where $\mathbf{u}_{\mathrm{c}}$ represents coarse unknowns, $\mathbf{u}_{\mathrm{r}}^i$ local interface unknowns that are not coarse, $\mathbf{I}_k$ is the identity matrix of dimension k and $\mathbf{S}_{\mathrm{rr}}^i$ is symmetric positive definite matrix of dimension n. Matrix $\mathbf{S}_{\mathrm{rr}}^i$ represents local Schur complement for i-th subdomain problem with zero values prescribed at coarse nodes. In the case of two subdomains, from (5) and (6) we have $\mathbf{S}_{\mathrm{rr}}^1 + \mathbf{S}_{\mathrm{rr}}^2 = \widehat{\mathbf{S}}_{\mathrm{rr}}$. From $\mathbf{E}\mathbf{R} = \mathbf{I}_{n+m}$ it follows

$$\mathbf{E} = \begin{bmatrix} \mathbf{A} & \mathbf{I}_n - \mathbf{A} & \mathbf{0} \\ \mathbf{C} & -\mathbf{C} & \mathbf{I}_m \end{bmatrix}, \tag{7}$$

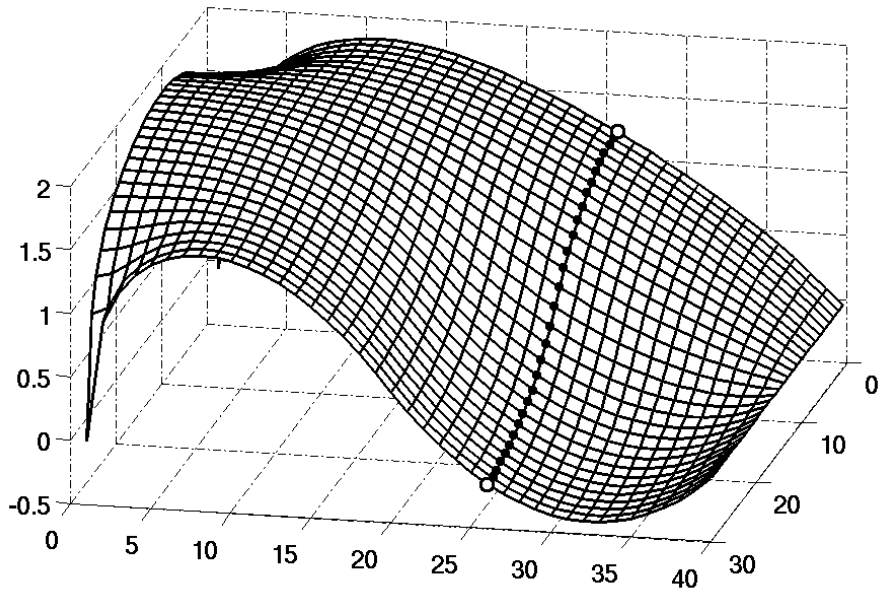

Fig. 1: *Test problem. 2D Poisson equation on a rectangular domain divided into two rectangular subdomains – left and right ones. Values of the solution at the interface nodes are marked by dots. The two coarse nodes are chosen on the opposite sides of the interface and are marked by circles.*

where $\mathbf{A}$ can be any weighting matrix for nodes that are not coarse and $\mathbf{C}$ is any matrix. Now we have the following decomposition of unity:

$$\mathbf{I} = \mathbf{RE} + (\mathbf{I} - \mathbf{RE}) = \begin{bmatrix} \mathbf{A} & \mathbf{I}_n - \mathbf{A} & \mathbf{0} \\ \mathbf{A} & \mathbf{I}_n - \mathbf{A} & \mathbf{0} \\ \mathbf{C} & -\mathbf{C} & \mathbf{I}_m \end{bmatrix} + \begin{bmatrix} \mathbf{I}_n - \mathbf{A} & \mathbf{A} - \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A} & \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{C} & \mathbf{0} \end{bmatrix} \tag{8}$$

(for brevity we write $\mathbf{I}$ instead of $\mathbf{I}_{2n+m}$). The projection $\mathbf{RE}$ can be viewed as some weighted average of values from adjacent subdomains at the interface nodes and the complementary projection $\mathbf{I} - \mathbf{RE}$ (which has the same energetic norm and is used in FETI-DP) as a weighted jump in these values. Its action on a given vector $\mathbf{u}$ of values at interface nodes can be expressed as

$$(\mathbf{I} - \mathbf{RE})\,\mathbf{u} = \begin{bmatrix} \mathbf{I}_n - \mathbf{A} & \mathbf{A} - \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A} & \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathrm{r}}^1 \\ \mathbf{u}_{\mathrm{r}}^2 \\ \mathbf{u}_{\mathrm{c}} \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{I}_n)\,\mathbf{d} \\ \mathbf{A}\,\mathbf{d} \\ \mathbf{C}\,\mathbf{d} \end{bmatrix}, \tag{9}$$

where $\mathbf{d} = \mathbf{u}_{\mathrm{r}}^2 - \mathbf{u}_{\mathrm{r}}^1$ is the jump in values at interface nodes that are not coarse.

For simplicity it is usually assumed that $\mathbf{C} = \mathbf{0}$ and $\mathbf{A}$ is diagonal. In what follows, we will try to achieve optimality only within this restricted class of choice of $\mathbf{E}$.

4.2 Approximate minimization of the energy norm of the projection

Our approach is to start with some fixed $\mathbf{u}$ with the interface jump $\mathbf{d}$ and try to find $\mathbf{E}$ so that it minimizes energetic difference between $\mathbf{u}$ and $\widehat{\mathbf{u}} = \mathbf{E}\mathbf{u}$. In other words, we are trying to minimize the energy norm of the projection $(\mathbf{I} - \mathbf{R}\mathbf{E})\mathbf{u}$ of the given vector $\mathbf{u}$. The square of the energy norm can be expressed as

$$||(\mathbf{I} - \mathbf{R}\mathbf{E})\mathbf{u}||_S^2 = \mathbf{u}^T(\mathbf{I} - \mathbf{R}\mathbf{E})^T\mathbf{S}(\mathbf{I} - \mathbf{R}\mathbf{E})\mathbf{u} = \mathbf{d}^T(\mathbf{A}^T\,\widehat{\mathbf{S}}_{\mathrm{rr}}\mathbf{A} - \mathbf{A}^T\mathbf{S}^1_{\mathrm{rr}} - \mathbf{S}^1_{\mathrm{rr}}\mathbf{A} + \mathbf{S}^1_{\mathrm{rr}}\,)\,\mathbf{d}\,.$$

Here we use the fact that $\widehat{\mathbf{S}}_{\mathrm{rr}} = \mathbf{S}^1_{\mathrm{rr}} + \mathbf{S}^2_{\mathrm{rr}}$ in the case of two subdomains.

A considerable effort is invested into minimization of this norm using the definition of $\mathbf{R}$ in the *adaptive BDDC* method [2]. Here, we follow a different path and concentrate on the matrix $\mathbf{E}$. Let $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Then the formula above can be seen as a quadratic function of variables α_i, which can be minimised by computing all partial derivatives and equating them to zero:

$$\frac{\partial}{\partial\alpha_i}||(\mathbf{I} - \mathbf{R}\mathbf{E})\mathbf{u}||_S^2 = 2d_i\left(\sum_j \widehat{s}_{ij}\alpha_j d_j - \sum_j s^1_{ij}d_j\right) = 0 \quad \forall\, i\,. \tag{10}$$

Here d_i stands for the i-th component of the *jump vector* $\mathbf{d}$, elements of the matrix $\widehat{\mathbf{S}}_{\mathrm{rr}}$ are denoted as $\widehat{s}_{ij}$, and elements of the matrix $\mathbf{S}^1_{\mathrm{rr}}$ are denoted as s^1_{ij}. Problem (10) represents solution of a system of linear equations of dimension n with a dense system matrix. Values of α_i obtained from (10) are tailored to the interface jump $\mathbf{d}$ of the given $\mathbf{u}$. Vector $\mathbf{u}$ changes in every iteration step, so values of α_i are also recomputed.

Solving (10) is in general nearly equally difficult as solving the original system (3). In order to solve this system with a reasonable effort, we can use some simplifying assumptions and solve it only approximately. One option is to omit all off-diagonal entries of matrices $\widehat{\mathbf{S}}_{\mathrm{rr}}$ and $\mathbf{S}^1_{\mathrm{rr}}$, which leads to the popular choice of

$$\alpha_i = s^1_{ii}/(s^1_{ii} + s^2_{ii})\,. \tag{11}$$

It is interesting to notice that in this case the solution does not depend on the chosen jump vector $\mathbf{d}$ and we can consider it as some approximation of minimising norm of the projection $\mathbf{I}-\mathbf{R}\mathbf{E}$ as a whole. The main drawback of this choice is the necessity of computing the values of the diagonal entries of the matrices $\mathbf{S}$ and $\widehat{\mathbf{S}}$, which otherwise need not be explicitly computed. For this reason, corresponding values at diagonal of original matrices $\mathbf{K}^1$ and $\mathbf{K}^2$ are often used instead of diagonal entries of Schur complements in formula (11) (e.g. in [1]).

4.3 A new construction of the averaging operator

We propose another approach. Let us assume $\mathbf{d}$ to be some test vector chosen so that it simplifies the system of equations (10). One option is to choose all the cartesian basis vectors $\mathbf{e}_k$, one after another – then we again arrive at solution (11).

For less elementary test vectors $\mathbf{d}$ we make an additional simplification: Let us assume that all α_i are equal to the same value of α for some set of nodes (so we

are going to find some average value). This is not as strange assumption as it may seem at the first glance: for large problems divided into a lot of relatively small subdomains by some automatic graph tool we probably can expect homogeneous behaviour along the interface for most pairs of adjacent subdomains. Then, after adding all equations (10) together, we get

$$\alpha = \mathbf{d}^T \mathbf{S}^1_{\mathrm{rr}}\, \mathbf{d} / \mathbf{d}^T (\mathbf{S}^1_{\mathrm{rr}} + \mathbf{S}^2_{\mathrm{rr}})\, \mathbf{d}. \tag{12}$$

This formula requires only matrix-vector products that are already computed in the PCG method and it can be generalized to more than 2 subdomains. Our proposition is to choose several test vectors with nonzero values at some selected nodes only (typically face or edge) and compute corresponding value of α.

5 Numerical results and conclusion

For a simple preliminary test depicted in Figure 1, a 2D Poisson equation on a rectangular domain was chosen. The domain was divided into two rectangular subdomains (different in size), both of which touch the boundary with prescribed Dirichlet boundary condition. The problem was discretized by FEM with bilinear elements. BDDC was used just as an iteration method, not as a preconditioner combined with PCG. Four different methods for choice of the averaging operator E were tested:

- I : arithmetic average, i.e. $\alpha = 0.5$,
- II : weighted average (11), i.e. $\alpha_i = s^1_{ii}/(s^1_{ii} + s^2_{ii})$,
- III : proposition (12) with $d = (1, \ldots, 1)$, i.e. $\alpha = \sum_{i,j} s^1_{ij} / \sum_{i,j} (s^1_{ij} + s^2_{ij})$,
- IV : proposition (12) with d chosen as actual interface jump.

Table 1 contains norms of errors (differences from exact solution) at first 5 iterations. There are two different choices of coarse unknowns: either none (first part of the table), or 2 nodes at the opposite ends of the interface (second part). For Method II, computed values of α_i were between 0.499 and 0.500 in both cases (i.e. very close to the arithmetic average). For Method III, value of α was 0.191 for the first case and 0.341 for the second. For Method IV, values of α were recomputed in every step and are presented in the last column.

For this simple test problem, it seems that Methods III and IV outperform Methods I and II. An interesting observation is that for the first three methods, involving coarse unknowns lead to better performance as one would expect, but in the case of Method IV the opposite is true, and although Method IV was absolutely excellent in the first case, with coarse nodes it worsens so that Method III becomes slightly better. These are just preliminary results and more general numerical tests will be performed for other 2D as well as 3D problems.

iter.	Method I	Method II	Method III	Method IV	α
without coarse nodes					
1.	1.7909	1.7851	0.9373	1.7909	0.500
2.	1.1010	1.0938	0.3034	0.0022	0.193
3.	0.6769	0.6702	0.0982	0.0004	0.273
4.	0.4161	0.4107	0.0318	7e-07	0.475
5.	0.2558	0.2517	0.0103	4e-11	0.191
2 coarse nodes					
1.	0.8663	0.8635	0.2690	0.8663	0.5
2.	0.2576	0.2560	0.0302	0.0476	0.316
3.	0.0766	0.0759	0.0035	0.0056	0.314
4.	0.0227	0.0225	0.0004	0.0007	0.314
5.	0.0068	0.0067	5e-05	8e-05	0.314

Tab. 1: *Comparison of discussed methods: errors at first 5 iterations for the test problem depicted in Figure 1, without (top) and with (bottom) coarse unknowns.*

References

[1] Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25** (2003), 246–258. doi: 10.1137/S1064827502412887.

[2] Mandel, J. and Sousedík, B.: Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. Comput. Methods Appl. Mech. Engrg. **196** (2007), 1389–1399.

[3] Mandel, J. and Sousedík, B.: BDDC and FETI-DP under minimalist assumptions. Computing **81** (2007), 269–280.